

Exhibit 9

Exhibit 10 – Belanovic and Leeser

'156 Patent

Claim Limitation (Claim 7)	Exemplary Disclosure
<p>[156a] A device comprising:</p>	<p>As reflected in Pavle Belanovic's thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i>, and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i>, they made and used a system that consists of a device. <i>See, e.g.:</i></p> <p>“The Wildstar Reconfigurable Computing Engine</p> <p>Reconfigurable computing is characterized by use of hardware elements that have reconfigurable architectures, as opposed to general purpose computing which uses hardware elements with fixed architectures.</p> <p>Many reconfigurable computing systems are based on one or more FPGAs connected to a number of memory banks. All designs presented in this project are implemented on the Wildstar reconfigurable computing engine from Annapolis Micro Systems. Figure 1.2 shows the structure of this board.</p> <p>Some of the main features of the Wildstar board are:</p> <ul style="list-style-type: none"> ● 3 Xilinx VIRTEX XCV1000 FPGAs, ● total of 3 million system gates, ● 40 Mbytes of SRAM, ● 1.6 Gbytes/sec I/O bandwidth, ● 6.4 Gbytes/sec memory bandwidth, ● processing clock rates up to 100MHz.” <p>Belanovic, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i> at 14.</p> <p>“For synthesis and mapping of all designs in this project we used Synplicity Pro from Synplify. Mapping, placing and routing of the designs was done using Xilinx Alliance tools. In order to verify the fidelity of the VHDL descriptions to the intended</p>

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	<p>functionality, all designs in this project were simulated with Mentor Graphics ModelSim prior to being implemented in hardware.” Belanovic, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i> at 13-14.</p>
<p>[156b] at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,</p>	<p>As reflected in Pavle Belanovic’s thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i>, and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i>, they made and used a system with at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. <i>See, e.g.:</i></p> <p>“Thus, module parameterized priority encoder has been developed, taking a signal to be examined on its input and producing the value, in unsigned fixed-point representation, of the index of the most significant ‘1’ in the input signal. The module is parameterized by the width of the input signal, as well as the width of the output signal.” Belanovic, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i> at 25-26.</p> <p>“The final stage of both the signed and the unsigned architectures is the output stage, where the computed fixed-point representation is placed on the output, unless the input was zero or an exception was encountered during operation or received at the input, in which case the output is set to all zeros.” Belanovic, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i> at 43.</p> <p><i>See also</i> Belanovic, <i>Library of Parameterized Hardware</i> at 15 (Fig 1.2) (showing 32-bit inputs and outputs to PE 1 and PE 2).</p>

Case 1:19-cv-12551-FDS Document 361-9 Filed 08/24/22 Page 4 of 23

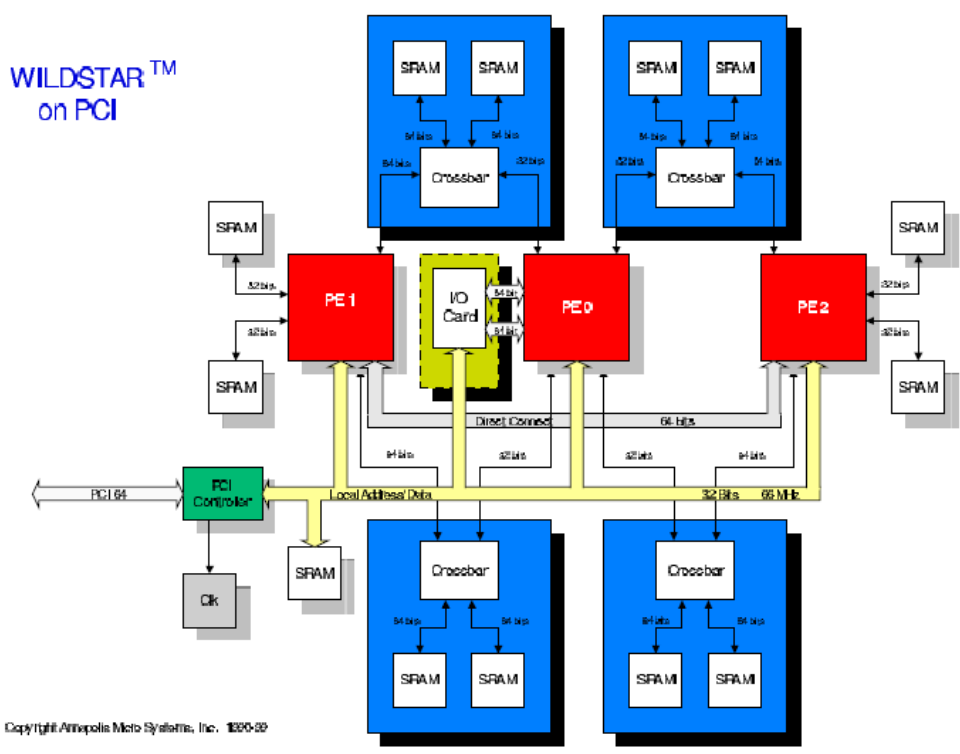
Claim Limitation (Claim 7)	Exemplary Disclosure
	 <p data-bbox="772 966 1035 984">Copyright Antipodia Micro Systems, Inc. 1990-99</p> <p data-bbox="804 1036 1694 1070">Figure 1.2: Structure of the Wildstar reconfigurable computing engine</p>
	<p>See also Belanovic, <i>Library of Parameterized Hardware</i> at 35 (Fig 2.4) (showing the inputs and outputs to the floating-point multiplier); see generally Belanovic, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i> at 35-43 (Section 2.4) (explaining hardware models for conversation).</p>

Exhibit 10 – Belanovic and Leeser

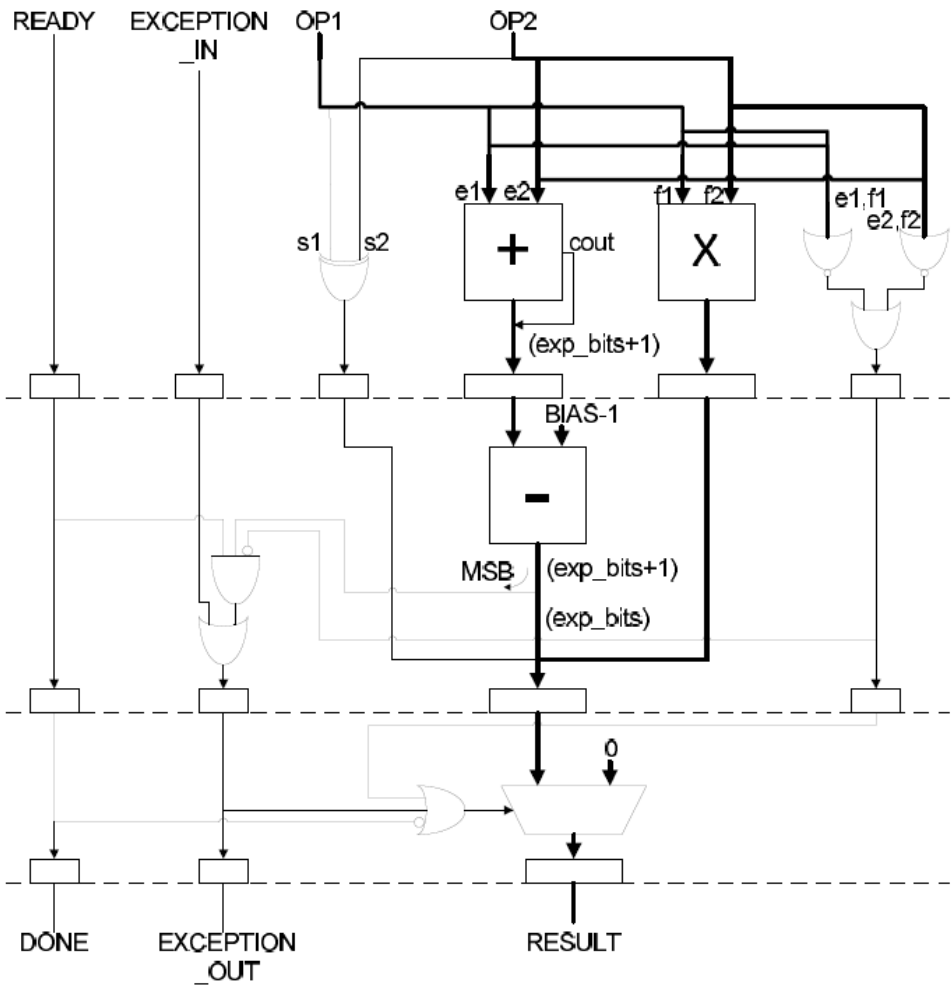
Claim Limitation (Claim 7)	Exemplary Disclosure
	 <p data-bbox="982 1274 1507 1307">Figure 2.4: Floating-point multiplication</p> <p data-bbox="693 1369 1747 1401">“We have developed a library of fully parameterized hardware modules for format</p>

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	<p>control, arithmetic operations and conversion to and from any fixed-point format. The format converters allow for hybrid implementations that combine both fixed and floating-point calculations. This permits the designer to choose between the increased range of floating-point and the increased precision of fixed-point within the same application.” Belanovic and Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> at 1.</p> <p>“A natural tradeoff exists between smaller bitwidths requiring fewer hardware resources and higher bitwidths providing better precision. Also, within a given total bitwidth, it is possible to assign various combinations of bitwidths to the exponent and fraction fields, where wider exponents result in higher range and wider fractions result in better precision. . . . Often, much smaller bitwidths than those specified in the 754 standard are sufficient to provide the desired precision. Reduced bitwidth implementations require fewer resources and thus allow for more parallel implementations than using the full IEEE standard format. In custom hardware designs, it is possible, and indeed desirable, to have full control and flexibility over the exact floating-point format implemented.” Belanovic and Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> at 2.</p> <p>As it relates to the Court’s construction of LPHDR execution unit, Belanovic’s thesis demonstrates that he and Miriam Leeser made and used a system with (1) addressable memory paired with the processing element(s); and (2) control for the processing elements (to the extent that control is interpreted to include any signaling that affects the operation of the processing element). See, e.g., Belanovic, <i>Library of Parameterized Hardware</i> at 16 (“Communication [between the FPGA] and the general purpose processor is done through memory banks...”), 20 (“All our modules are equipped with ready and done pipeline synchronization signals,...”), 31-32 Fig. 2.3 (describing, inter alia, use of “EXCEPTION_IN” signal to affect operation of floating-point adder).</p> <p>To the extent Singular contends that Belanovic’s thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i>, and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules</i></p>

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	<p>and Their Use, do not themselves demonstrate that Belanovic and Leeser made and used a system with addressable memory paired with the processing element(s) and control for the processing elements, FPGA-based parallel processors that had addressable memory paired with the processing element(s) and control for the processing elements (to the extent that control is interpreted to include any signaling that affects the operation of the processing element) were well-known in the art, as explained in Section IV.C.1.d of the Amended Responsive Contentions Regarding Non-Infringement and Invalidity. See, e.g., Shirazi, at 160-161. To the extent Singular nonetheless contends that one of skill in the art would have needed a motivation to configure FPGA processing elements with paired addressable memory and/or control, one of skill in the art would have been motivated to do so based on the teachings of any of Dockser, Shirazi, Cray T3D, Linehart, the admitted prior art, Patterson & Hennessy, in Computer Organization & Design, The Hardware Software Interface (3d. Ed. 2005), and/or Hamada.</p>
<p>[156c] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; and</p>	<p>As reflected in Pavle Belanovic's thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i>, and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i>, they made and used a system with the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. See, e.g.:</p> <p>“The floating-point formats in our work are a generalized superset of all these formats. It includes all the IEEE formats as particular instances of exponent and mantissa bitwidths, as well as the flexible floating-point format presented by Dido et al.[2] and the two formats by Shirazi et al.[17].” Belanovic, <i>Library of Parameterized Hardware</i> at 19.</p> <p>“The experiments were conducted by synthesizing the modules for specific floating-</p>

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure																																																																																																																																							
	<p>point formats on the Annapolis Micro Systems Wildstar reconfigurable computing engine (see Section 1.3). Table 2.2 shows results of the synthesis experiments on floating-point operator modules. The quantities for the area of each instance are expressed in slices of the Xilinx XCV1000 FPGA. Results for the fp_add module in Table 2.2 also represent the fp_sub module, which has the same amount of logic.</p> <p>Floating-point formats used in the experiments were chosen to represent the range of realistic floating-point formats from 8 to 32 bits in total bitwidth and include the IEEE single precision format (E1 in Table 2.2).” Belanovic, <i>Library of Parameterized Hardware</i> at 46-47.</p> <div><div>Table 2.2: Operator synthesis results</div><table><tr><th rowspan="2">Format</th><th colspan="3">Bitwidth</th><th colspan="2">Area</th><th colspan="2">Per IC</th></tr><tr><th>total</th><th>exponent</th><th>fraction</th><th>fp_add</th><th>fp_mul</th><th>fp_add</th><th>fp_mul</th></tr><tr><td>A0</td><td>8</td><td>2</td><td>5</td><td>39</td><td>46</td><td>236</td><td>200</td></tr><tr><td>A1</td><td>8</td><td>3</td><td>4</td><td>39</td><td>51</td><td>236</td><td>180</td></tr><tr><td>A2</td><td>8</td><td>4</td><td>3</td><td>32</td><td>36</td><td>288</td><td>256</td></tr><tr><td>B0</td><td>12</td><td>3</td><td>8</td><td>84</td><td>127</td><td>109</td><td>72</td></tr><tr><td>B1</td><td>12</td><td>4</td><td>7</td><td>80</td><td>140</td><td>115</td><td>65</td></tr><tr><td>B2</td><td>12</td><td>5</td><td>6</td><td>81</td><td>108</td><td>113</td><td>85</td></tr><tr><td>C0</td><td>16</td><td>4</td><td>11</td><td>121</td><td>208</td><td>76</td><td>44</td></tr><tr><td>C1</td><td>16</td><td>5</td><td>10</td><td>141</td><td>178</td><td>65</td><td>51</td></tr><tr><td>C2</td><td>16</td><td>6</td><td>9</td><td>113</td><td>150</td><td>81</td><td>61</td></tr><tr><td>D0</td><td>24</td><td>6</td><td>17</td><td>221</td><td>421</td><td>41</td><td>21</td></tr><tr><td>D1</td><td>24</td><td>8</td><td>15</td><td>216</td><td>431</td><td>42</td><td>21</td></tr><tr><td>D2</td><td>24</td><td>10</td><td>13</td><td>217</td><td>275</td><td>42</td><td>33</td></tr><tr><td>E0</td><td>32</td><td>5</td><td>26</td><td>328</td><td>766</td><td>28</td><td>12</td></tr><tr><td>E1</td><td>32</td><td>8</td><td>23</td><td>291</td><td>674</td><td>31</td><td>13</td></tr><tr><td>E2</td><td>32</td><td>11</td><td>20</td><td>284</td><td>536</td><td>32</td><td>17</td></tr></table></div> <p>“The hardware modules described in Chapter 2 lend themselves to the creation of finely customized hardware implementations of algorithms. They give the designer full freedom to implement various sections of the algorithm in the most suitable arithmetic representation, be it fixed or floating-point. Also, bitwidths of all the signals in the circuit, whether in fixed or floating point representation, can be optimized to the</p>	Format	Bitwidth			Area		Per IC		total	exponent	fraction	fp_add	fp_mul	fp_add	fp_mul	A0	8	2	5	39	46	236	200	A1	8	3	4	39	51	236	180	A2	8	4	3	32	36	288	256	B0	12	3	8	84	127	109	72	B1	12	4	7	80	140	115	65	B2	12	5	6	81	108	113	85	C0	16	4	11	121	208	76	44	C1	16	5	10	141	178	65	51	C2	16	6	9	113	150	81	61	D0	24	6	17	221	421	41	21	D1	24	8	15	216	431	42	21	D2	24	10	13	217	275	42	33	E0	32	5	26	328	766	28	12	E1	32	8	23	291	674	31	13	E2	32	11	20	284	536	32	17
Format	Bitwidth			Area		Per IC																																																																																																																																		
	total	exponent	fraction	fp_add	fp_mul	fp_add	fp_mul																																																																																																																																	
A0	8	2	5	39	46	236	200																																																																																																																																	
A1	8	3	4	39	51	236	180																																																																																																																																	
A2	8	4	3	32	36	288	256																																																																																																																																	
B0	12	3	8	84	127	109	72																																																																																																																																	
B1	12	4	7	80	140	115	65																																																																																																																																	
B2	12	5	6	81	108	113	85																																																																																																																																	
C0	16	4	11	121	208	76	44																																																																																																																																	
C1	16	5	10	141	178	65	51																																																																																																																																	
C2	16	6	9	113	150	81	61																																																																																																																																	
D0	24	6	17	221	421	41	21																																																																																																																																	
D1	24	8	15	216	431	42	21																																																																																																																																	
D2	24	10	13	217	275	42	33																																																																																																																																	
E0	32	5	26	328	766	28	12																																																																																																																																	
E1	32	8	23	291	674	31	13																																																																																																																																	
E2	32	11	20	284	536	32	17																																																																																																																																	

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	<p>precision required by the values the signal carries.</p> <p>When using floating-point arithmetic, the designer using the library has full control to trade off between range and precision. Because all the modules in the library are fully parameterized, the boundary between the exponent and fraction fields for the same total bitwidth is flexible. With a wider exponent field, the designer provides larger range to the signal, while sacrificing precision. Similarly, to increase the precision of a signal at the cost of reduced range, the designer chooses a narrower exponent and wider fraction field.” Belanovic, <i>Library of Parameterized Hardware</i> at 50. Thus, the floating point library was capable of implementing various formats for any given bitwidth, and was not limited to the fraction and exponent combinations set forth in Table 2.2 of the Belanovic thesis. The formats implemented by the library could have included, for example, the floating point format, including the exponent and fraction widths described in the following: Sudha et al., <i>An Efficient Digital Architecture for Principal Component Neural Network and its FPGA Implementation</i> (2007) at 428 (each parameter in the design is represented as a floating point number with an 8-bit mantissa and an 8-bit exponent); Tong et al., <i>Reducing Power by Optimizing the Necessary Precision/Range of Floating-Point Arithmetic</i> (2000) at 273 (each parameter in the design is represented as a floating point number with a 5-bit fraction and 6-bit exponent); Texas Instruments TMS320C32 DSP (1995) (each parameter in the design is represented as a floating point number with a 7-bit fraction and 8-bit exponent); and Cray T3D System (1994) (each parameter in the design is represented as a floating point number with a 5-bit fraction and 11-bit exponent).</p> <p>“This line of thought was expanded on by the significant work of Shirazi et al. [17] who suggested application-specific formats for image and DSP algorithms in widths of 16 (1-6-9) and 18 (1-7-10) bits, as opposed to the full 32 (1-8-23) bits in the IEEE standard.” Belanovic, <i>Library of Parameterized Hardware</i> at 18.</p> <p>“Results for the fp add module in Table 2 also represent the fp sub module, which has the same amount of logic. The results in Table 2 show growth in area with increasing total bitwidth, for both modules.” Belanovic and Leeser, <i>Library of Parameterized</i></p>

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure																																																																																																						
	<p><i>Floating-Point Modules and Their Use</i> at 7.</p> <p style="text-align: center;">Table 2. Operator synthesis results</p> <table><tr><th>Format</th><th>Total bits</th><th>Exponent</th><th>Fraction</th><th colspan="2">Area</th></tr><tr><td></td><td></td><td></td><td></td><th>fp_add</th><th>fp_mul</th></tr><tr><td>A0</td><td>8</td><td>2</td><td>5</td><td>39</td><td>46</td></tr><tr><td>A1</td><td>8</td><td>3</td><td>4</td><td>39</td><td>51</td></tr><tr><td>A2</td><td>8</td><td>4</td><td>3</td><td>32</td><td>36</td></tr><tr><td>B0</td><td>12</td><td>3</td><td>8</td><td>84</td><td>127</td></tr><tr><td>B1</td><td>12</td><td>4</td><td>7</td><td>80</td><td>140</td></tr><tr><td>B2</td><td>12</td><td>5</td><td>6</td><td>81</td><td>108</td></tr><tr><td>C0</td><td>16</td><td>4</td><td>11</td><td>121</td><td>208</td></tr><tr><td>C1</td><td>16</td><td>5</td><td>10</td><td>141</td><td>178</td></tr><tr><td>C2</td><td>16</td><td>6</td><td>9</td><td>113</td><td>150</td></tr><tr><td>D0</td><td>24</td><td>6</td><td>17</td><td>221</td><td>421</td></tr><tr><td>D1</td><td>24</td><td>8</td><td>15</td><td>216</td><td>431</td></tr><tr><td>D2</td><td>24</td><td>10</td><td>13</td><td>217</td><td>275</td></tr><tr><td>E0</td><td>32</td><td>5</td><td>26</td><td>328</td><td>766</td></tr><tr><td>E1</td><td>32</td><td>8</td><td>23</td><td>291</td><td>674</td></tr><tr><td>E2</td><td>32</td><td>11</td><td>20</td><td>284</td><td>536</td></tr></table> <p>“The algorithm has been partitioned so the distance calculation is performed in floating-point arithmetic with 5 exponent and 6 fraction bits (1-5-6 format), while the comparison and accumulation operations are performed in 12-bit unsigned fixed-point format. Input data is in 12-bit unsigned fixed-point format, so it needs to be converted to the 1-5-6 floating-point representation.” Belanovic and Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> at 8-9.</p>	Format	Total bits	Exponent	Fraction	Area						fp_add	fp_mul	A0	8	2	5	39	46	A1	8	3	4	39	51	A2	8	4	3	32	36	B0	12	3	8	84	127	B1	12	4	7	80	140	B2	12	5	6	81	108	C0	16	4	11	121	208	C1	16	5	10	141	178	C2	16	6	9	113	150	D0	24	6	17	221	421	D1	24	8	15	216	431	D2	24	10	13	217	275	E0	32	5	26	328	766	E1	32	8	23	291	674	E2	32	11	20	284	536
Format	Total bits	Exponent	Fraction	Area																																																																																																			
				fp_add	fp_mul																																																																																																		
A0	8	2	5	39	46																																																																																																		
A1	8	3	4	39	51																																																																																																		
A2	8	4	3	32	36																																																																																																		
B0	12	3	8	84	127																																																																																																		
B1	12	4	7	80	140																																																																																																		
B2	12	5	6	81	108																																																																																																		
C0	16	4	11	121	208																																																																																																		
C1	16	5	10	141	178																																																																																																		
C2	16	6	9	113	150																																																																																																		
D0	24	6	17	221	421																																																																																																		
D1	24	8	15	216	431																																																																																																		
D2	24	10	13	217	275																																																																																																		
E0	32	5	26	328	766																																																																																																		
E1	32	8	23	291	674																																																																																																		
E2	32	11	20	284	536																																																																																																		

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	<p>“Also, all bitwidths in the datapath, whether in fixed or floating-point representation, can be optimized to the precision required for that signal. Hence, designing with our library of parameterized modules avoids expensive inefficiencies that are inherent in designs that operate only in the IEEE standard formats. In fact, such inefficiencies occur in any design that is restricted to using a small set of particular formats, even if these are custom. Using our library of parameterized modules provides the finest-grain control possible over datapath bitwidths. Finally, when using floating-point arithmetic, the designer has full control to trade off between range and precision. Because our modules are fully parameterized, the boundary between the exponent and the fraction for the same total bitwidth is flexible. Thus, with a wider exponent field, the designer has larger range for a value while sacrificing precision. Similarly, to increase the precision of a signal at the cost of reduced range, the designer chooses a narrower exponent and wider fraction field.” Belanovic and Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> at 8-9.</p> <p>To the extent that Singular contends that this system does not itself identify a floating-point format that meets the particular range and error requirements, notwithstanding its disclosure of a floating point format with 9 fraction bits and 6 exponent bits, that format would have been obvious to one of skill in the art for the reasons explained in the Responsive Contentions Regarding Non-Infringement and Invalidity, which explain how those skilled in the art would mix and match formats depending on application specific needs. <i>See also</i> Belanovic, <i>Library of Parameterized Hardware</i> at 16 (explaining how custom datapaths for fixed- and floating-point arithmetic would have “optimal signal bitwidths throughout the custom datapath [that] are application-specific and depend on the values they carry.”) For example, one of skill in the art would have understood the different combinations of fraction and exponent bits (<i>e.g.</i>, 5 fraction bits, 6 exponent bits, and one sign bit, for a total of 12 bits) would have been possible and even desired depending on the application. Alternatively, one of skill in the art would have been motivated to apply the teachings of Tong, which included a 5-bit mantissa and 6-bit exponent (<i>see</i> Tong chart) because Tong is cited. <i>See</i> Belanovic, <i>Library of Parameterized Hardware</i> at 73, n.21.</p>

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	<p><i>See also</i> Appendix to Responsive Contentions Regarding Non-Infringement and Invalidity (detailing error rates associated with different mantissa sizes).</p>
<p>[156d] at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit;</p>	<p>As reflected in Pavle Belanovic’s thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i>, and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i>, they made and used a system with at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit. <i>See, e.g.:</i></p> <p>“The Wildstar Reconfigurable Computing Engine</p> <p>Reconfigurable computing is characterized by use of hardware elements that have reconfigurable architectures, as opposed to general purpose computing which uses hardware elements with fixed architectures.</p> <p>Many reconfigurable computing systems are based on one or more FPGAs connected to a number of memory banks. All designs presented in this project are implemented on the Wildstar reconfigurable computing engine from Annapolis Micro Systems. Figure 1.2 shows the structure of this board.</p> <p>Some of the main features of the Wildstar board are:</p> <ul style="list-style-type: none"> ● 3 Xilinx VIRTEX XCV1000 FPGAs, ● total of 3 million system gates, ● 40 Mbytes of SRAM, ● 1.6 Gbytes/sec I/O bandwidth, ● 6.4 Gbytes/sec memory bandwidth, ● processing clock rates up to 100MHz.” <p>Belanovic, <i>Library of Parameterized Hardware</i> at 14.</p>

Exhibit 10 – Belanovic and Leeser

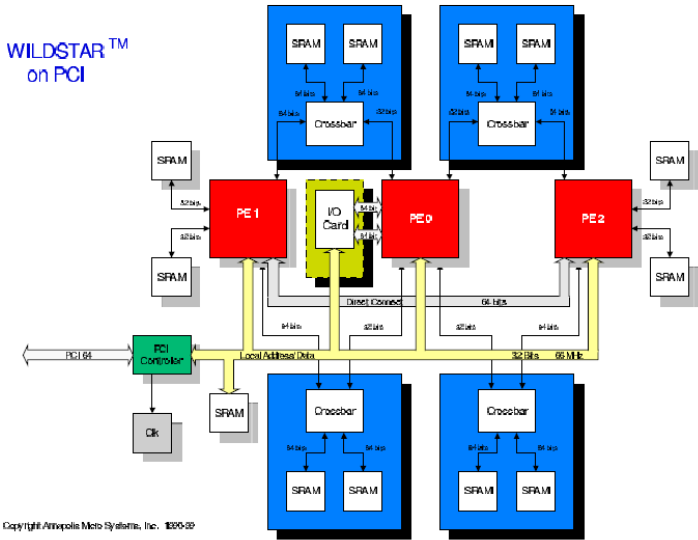
Claim Limitation (Claim 7)	Exemplary Disclosure
	 <p>Figure 1.2: Structure of the Wildstar reconfigurable computing engine</p>
<p>[156e] wherein the at least one first computing device comprises at least one of a central processing unit (CPU), a graphics processing unit (GPU), a field programmable gate array (FPGA), a microcode-based processor, a hardware sequencer, and a state machine;</p>	<p>As reflected in Pavle Belanovic’s thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i>, and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i>, they made and used a system with at least one first computing device comprises at least one of a central processing unit (CPU), a graphics processing unit (GPU), a field programmable gate array (FPGA), a microcode-based processor, a hardware sequencer, and a state machine. Specifically, Belanovic discloses a “host” computer that comprises at least a “state machine,” “FPGAs,” and various “processing units.” <i>See, e.g.,</i>:</p> <p>Belanovic, <i>Library of Parameterized Hardware</i> at 15 (Fig 1.2) (depicting the Wildstar computer engine, including (1) a PCI 64, which implies an interconnection with a PC and thus a CPU, and (2) an I/O Card and Controller, which qualifies as a state machine).</p>

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	<div data-bbox="793 228 1724 943"> <p>Figure 1.2: Structure of the Wildstar reconfigurable computing engine</p> </div> <p>“Field Programmable Gate Arrays (FPGAs) are integrated circuits with a flexible architecture, such that their structure can be programmed by the designer. FPGAs are composed of an array of hardware resources called configurable logic blocks (CLBs). The designer creates the functionality of the overall circuit by configuring CLBs to perform appropriate logic functions. Hence, FPGAs are a form of reconfigurable hardware, combining flexibility similar to software with the speed of specialized hardware.” Belanovic, <i>Library of Parameterized Hardware</i> at 13.</p> <p>“Typically, the parts of the algorithm that are assigned to software are serial or procedural in nature, while the highly parallel, computational parts of the algorithm get implemented in hardware. Custom datapaths are created in reconfigurable hardware to</p>

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	<p>achieve desired functionality. Communication with the general purpose processor is done through memory banks and/or register tables in reconfigurable hardware, both of which are accessible by the custom hardware and the general purpose processor.” Belanovic, <i>Library of Parameterized Hardware</i> at 15-16.</p>
<p>[156f] and, wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.</p>	<p>As reflected in Pavle Belanovic’s thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i>, and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i>, they made and used a system in which the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide. <i>See, e.g.:</i></p> <p>“Unlike in fixed-point, in floating-point arithmetic multiplication is a relatively straightforward operation compared to addition. This is due to the sign-exponent-magnitude nature of the floating-point format. The sign of the product is the exclusive OR (XOR) of the operand signs. The exponent of the product is the sum of the operand exponents. The mantissa is the product of the operand mantissas. Note that the operations on all three fields of the floating-point format are independent and can be implemented in parallel. The structure of the floating-point multiplier is shown in Figure 2.” Belanovic and Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> at 4-5.</p>

Exhibit 10 – Belanovic and Leeser

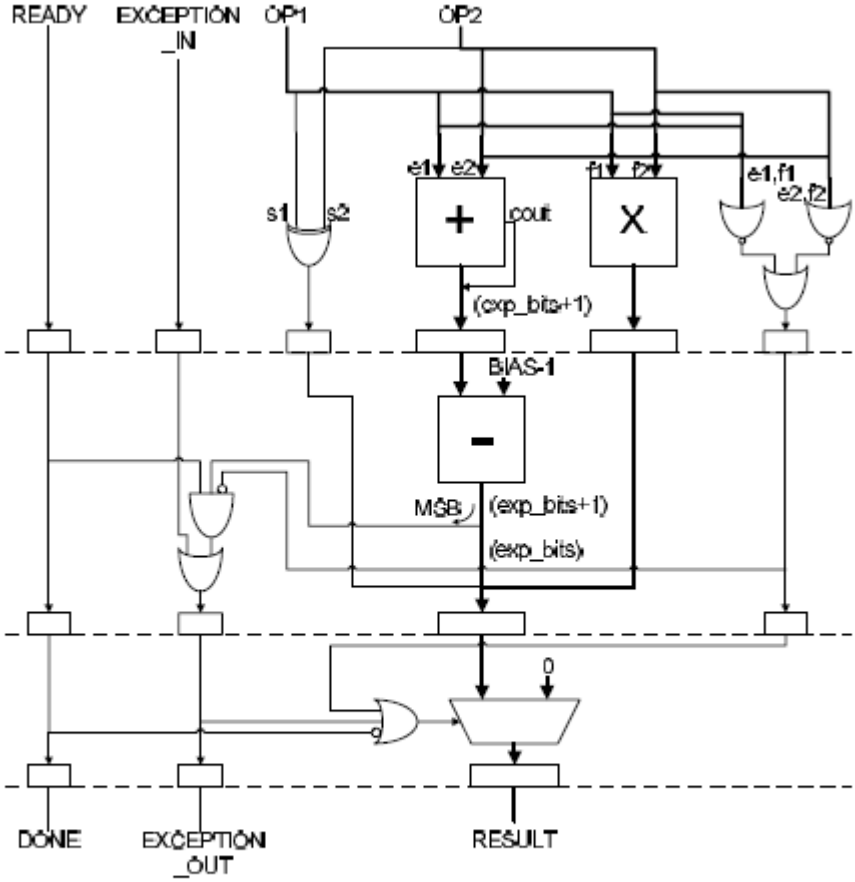
Claim Limitation (Claim 7)	Exemplary Disclosure
	 <p>The diagram illustrates a floating-point multiplication unit. It features two main data paths: one for addition (+) and one for multiplication (X). The inputs are READY, EXCEPTION_IN, OP1, and OP2. OP1 and OP2 are multiplexed to select between addition and multiplication. The diagram shows the internal logic for handling exponents and mantissas, including normalization and bias adjustment. The output is RESULT, with DONE and EXCEPTION_OUT signals indicating the completion of the operation.</p> <p>Fig. 2. Floating-point multiplication</p> <p>“Synthesis results indicate that a realistic design on a Xilinx XCV1000 FPGA may include up to 31 addition or 13 multiplication operators, complete with denormalizing,</p>

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	<p>rounding and normalizing functionality each, for the IEEE single precision format.</p> <p>Similarly, a useful custom floating-point format, with 5 exponent and 6 mantissa bits for example, may provide the designer with up to 113 addition or 85 multiplication modules, all also complete with full format handling functionalities, on the same FPGA.” Belanovic, <i>Library of Parameterized Hardware</i> at 34.</p> <p>“Some of the main features of the Wildstar board are: . . . 3 Xilinx VIRTEX XCV1000 FPGAs.” Belanovic, <i>Library of Parameterized Hardware</i> at 14.</p> <p>“All these modules are specified in VHDL and implemented on the Wildstar reconfigurable computing engine from Annapolis Microsystems, using a Xilinx XCV1000 FPGA. Synthesis results for parameterized arithmetic operator modules are presented in Table 2, for a set of floating-point formats labeled A0 through E2.” Belanovic and Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> at 7.</p> <p>Thus, the system described by Belanovic and Leeser could have a total of 255 LPHDR execution units (85 multiplication modules per each of three FPGAs).</p> <p>“[I]t can be concluded that the three fields of the floating-point format do not interact during multiplication and can thus be processed at the same time, in parallel. The sign of the product is given as the exclusive OR (XOR) of the input value signs. Mantissa of the product is calculated by fixed-point multiplication of the input value mantissas, while the exponents of the input values are added to give the exponent of the product.</p> <p>To the extent that Singular contends that Belanovic’s thesis does not identify a device with at least 100 multiplication execution units, notwithstanding its disclosure of a system with 255 such units, such a device would have been obvious given intervening FPGA developments for the reasons explained in the Responsive Contentions Regarding Non-Infringement and Invalidity.</p>

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 7)	Exemplary Disclosure
	<p>“Often, much smaller bitwidths than those specified in the 754 standard are sufficient to provide the desired precision. Reduced bitwidth implementations require fewer resources and thus allow for more parallel implementations than using the full IEEE standard format. In custom hardware designs, it is possible, and indeed desirable, to have full control and flexibility over the exact floating-point format implemented.” Belanovic and Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> at 2.</p> <p>“Each component is parameterized by exponent and mantissa bitwidths. Each component has a ready and a done signal to allow them to be easily assembled into larger designs.” Belanovic and Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> at 3.</p>

Exhibit 10 – Belanovic and Leeser

'273 Patent

Claim Limitation (Claim 53)	Exemplary Disclosure
[273a] A device:	As reflected in Pavle Belanovic's thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i> , and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> , they made and used a system that consists of a device. See [156a]
[273b] comprising at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,	As reflected in Pavle Belanovic's thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i> , and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> , they made and used a system with at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. See [156b].
[273c] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X % of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values	<p>As reflected in Pavle Belanovic's thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i>, and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i>, they made and used a system with the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. See [156c].</p> <p>To the extent that Singular contends that this system does not itself identify a floating-point format that meets the particular range and error requirements, notwithstanding its disclosure of a floating point format with 9 fraction bits and 6 exponent bits, that format would have been obvious to one of skill in the art for the</p>

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 53)	Exemplary Disclosure
of that same input;	reasons explained in the Responsive Contentions Regarding Non-Infringement and Invalidity.
[273d] wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.	As reflected in Pavle Belanovic's thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i> , and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> , they made and used a system in which the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide. <i>See</i> [156f].

Exhibit 10 – Belanovic and Leeser

'961 Patent

Claim Limitation (Claim 4)	Exemplary Disclosure
[961a] A device comprising:	As reflected in Pavle Belanovic's thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i> , and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> , they made and used a system that consists of a device. See [156a].
[961b] at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,	As reflected in Pavle Belanovic's thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i> , and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> , they made and used a system with at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. See [156b].
[961c] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=10% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.2% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; and	<p>As reflected in Pavle Belanovic's thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i>, and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i>, they made and used a system with the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. See [156c].</p> <p>To the extent that Singular contends that this system does not itself identify a floating-point format that meets the particular range and error requirements, notwithstanding its disclosure of a floating point format with 9 fraction bits and</p>

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 4)	Exemplary Disclosure
	6 exponent bits, that format would have been obvious to one of skill in the art for the reasons explained in the Responsive Contentions Regarding Non-Infringement and Invalidity.
[961d] at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit.	As reflected in Pavle Belanovic's thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i> , and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> , they made and used a system with at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit. <i>See</i> [156d].

Claim Limitation (Claim 13)	Exemplary Disclosure
[961e] A device comprising:	As reflected in Pavle Belanovic's thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i> , and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> , they made and used a system that consists of a device. <i>See</i> [156a].
[961f] a plurality of components comprising:	As reflected in Pavle Belanovic's thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i> , and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> , they made and used a system with at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit. <i>See above</i> [156d].
[961g] at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical	As reflected in Pavle Belanovic's thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i> , and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i> , they made and used a system with at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first input signal representing a first numerical value to produce a first output

Exhibit 10 – Belanovic and Leeser

Claim Limitation (Claim 13)	Exemplary Disclosure
value,	signal representing a second numerical value. <i>See</i> [156b].
<p>[961h] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=10% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.2% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input.</p>	<p>As reflected in Pavle Belanovic's thesis, <i>Library of Parameterized Hardware Modules for Floating-Point Arithmetic with an Example Application</i>, and his related article with Miriam Leeser, <i>Library of Parameterized Floating-Point Modules and Their Use</i>, they made and used a system with the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. <i>See</i> [156c].</p> <p>To the extent that Singular contends that this system does not itself identify a floating-point format that meets the particular range and error requirements, notwithstanding its disclosure of a floating point format with 9 fraction bits and 6 exponent bits, that format would have been obvious to one of skill in the art for the reasons explained in the Responsive Contentions Regarding Non-Infringement and Invalidity.</p>